Balancing Energy Preservation and Performance in Energy-Harvesting Sensor Networks

Hribar, Jernej; Shinkuma, Ryoichi; Akiyama, Kuon; Iosifidis, George; Dusparic, Ivana

DOI
10.1109/JSEN.2024.3469539

Publication date
2024

Document Version
Final published version

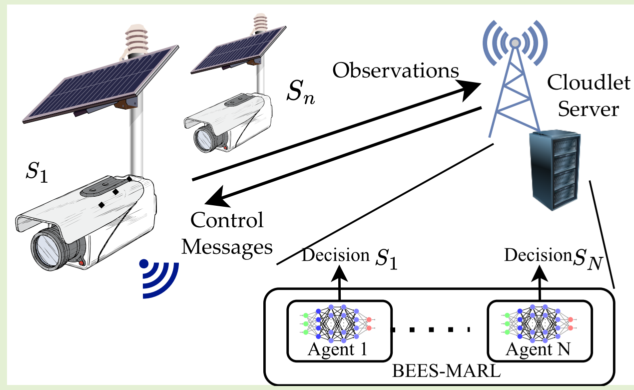
Published in
IEEE Sensors Journal

Citation (APA)
Hribar, J., Shinkuma, R., Akiyama, K., Iosifidis, G., & Dusparic, I. (2024). Balancing Energy Preservation and Performance in Energy-Harvesting Sensor Networks. *IEEE Sensors Journal*, *24*(22), 38352-38364. https://doi.org/10.1109/JSEN.2024.3469539

Balancing Energy Preservation and Performance in Energy-Harvesting Sensor Networks

Jernej Hribar, *Member, IEEE*, Ryoichi Shinkuma, *Senior Member, IEEE*, Kuon Akiyama, George Iosifidis, and Ivana Dusparic

***Abstract*—The development of environmentally friendly, green communications is at the forefront of designing future Internet of Things (IoT) networks, although many opportunities to improve energy conservation from energy-harvesting (EH) sensors remain unexplored. Ubiquitous computing power, available in the form of cloudlets, enables the processing of the collected observations at the network edge. Often, the information that the Artificial Intelligence of Things (AIoT) application obtains by processing observations from one sensor can also be obtained by processing observations from another sensor. Consequently, a sensor can take advantage of the correlation between processed observations to avoid unnecessary transmissions and save energy. For example, when two cameras monitoring the same intersection detect the same vehicles, the system can recognize this overlap and reduce redundant data transmissions. This approach allows the network to conserve energy while still ensuring accurate vehicle detection, thereby maintaining the overall performance of the AIoT task. In this article, we consider such a system and develop a novel solution named balancing energy efficiency in sensor networks with multiagent reinforcement learning (BEES-MARL). Our proposed solution is capable of taking advantage of correlations in a system with multiple EH-powered sensors observing the same scene and transmitting their observations to a cloudlet. We evaluate the proposed solution in two data-driven use cases to verify its benefits and in a general setting to demonstrate scalability. Our solution improves task performance, measured by recall, by up to 16% over a heuristic approach, while minimizing latency and preventing outages.**



***Index Terms*—Artificial Intelligence of Things (AIoT), deep learning (DL), edge computing, energy harvesting (EH), green communications, multiagent reinforcement learning (MARL).**

I. Introduction

RECENT advances in machine learning (ML) methods combined with the unprecedented increase in available computing power at the edge of the Internet of Things (IoT) network have opened up new opportunities for processing data collected by sensors [1]. Often, observations from multiple sensors are processed simultaneously by a resource-intensive Artificial Intelligence of Things (AIoT) task. For example, a cloudlet server [2] can process images from cameras to detect objects such as vehicles or anomalies such as traffic congestion. Moreover, in many such cases, the network can obtain the same information, for example, detect all vehicles in observed intersections, by processing observations from only a subset of available sensors. Consequently, the IoT network can benefit from utilizing such correlation in the information that can be extracted from the raw observations. In this article, we explore how sensor resources, in this case energy, can be conserved by utilizing correlation to forego acquisition and transmission, without compromising the performance of the AIoT task.

The larger and denser the IoT network, the more advantageous it becomes to take advantage of correlated information [3], [4]. For example, sensors monitoring overlapping areas can observe the same anomaly. At the same time, sensors are often limited in terms of energy and available

Received 6 September 2024; accepted 25 September 2024. Date of publication 3 October 2024; date of current version 14 November 2024. This work was supported in part by European Regional Development Fund through the Science Foundation Ireland (SFI) Research Centres Program under Grant 13/RC/2077_P2, in part by the SFI-NSFC Partnership Program through SFI CONNECT under Grant 17/NSFC/5224, in part by the Japan Science and Technology Agency (JST) PRESTO under Grant JPMJPR1854, in part by the Japan Society for Promotion of Science (JSPS) KAKENHI under Grant JP21H03427, in part by JSPS International Research Fellow under Grant PE20723, and in part by the Slovenian Research Agency under Grant P2-0016. The associate editor coordinating the review of this article and approving it for publication was Dr. Zhenghua Chen. *(Corresponding author: Jernej Hribar.)*

Jernej Hribar is with the Department of Communication Systems, Jožef Stefan Institute, 1000 Ljubljana, Slovenia (e-mail: jernej.hribar@ijs.si).

Ryoichi Shinkuma and Kuon Akiyama are with the Faculty of Engineering, Shibaura Institute of Technology, Tokyo 135-8548, Japan.

George Iosifidis is with the Department of Software Technology, Delft University of Technology, 2628 CD Delft, The Netherlands.

Ivana Dusparic is with CONNECT, Trinity College Dublin, Dublin, D02 PN40 Ireland.

Digital Object Identifier 10.1109/JSEN.2024.3469539

processing power. Therefore, a solution that takes advantage of the correlation in processed observations can help the network preserve limited resources, for example, bandwidth, energy, or processing power, and thus become more sustainable.

The environmentally friendly provision of the energy required for the operation of edge sensors, that is, the application of green communication practices, is an increasing challenge [5], [6], [7]. While connecting sensors to the power grid is the most reliable way to supply power, it also leads to high deployment and environmental costs. On the one hand, an energy harvesting (EH) proves to be a viable alternative as it allows the sensor to collect energy by tapping into various environmental sources such as wind, solar radiation, or vibration [8]. On the other hand, a sensor with an EH must be aware of its energy consumption while ensuring that the application delivers the desired performance. In addition, the energy collected by an EH varies over time. For example, a sensor with a solar panel may collect a lot of energy during the day, while the energy collected at night is negligible. In the system under study, a sensor powered by an EH can take advantage of the correlation to optimize energy conservation. For example, a sensor may decide to take no action and instead conserve energy because another sensor will obtain the required information. To develop a solution that can adapt to ever-changing conditions, we propose a deep reinforcement learning (DRL) approach that can learn how to take advantage of available energy and correlation.

Existing literature addresses only a single aspect of the investigated problem at the time, that is, either energy-efficient offloading or the performance of data-analytic tasks, but not both simultaneously. For example, Zhang and Chen [9], Min et al. [10], Balasubramanian et al. [11], Liu et al. [12], and Xu et al. [13] considered offloading computational tasks from an EH-powered device to a server, but without considering the performance of the task. Lyu et al. [14], Ran et al. [15], and Galanopoulos et al. [16], on the other hand, considered processing data, but do not consider EH. In addition, neither of the above approaches considers taking advantage of correlation as we propose in our work.

In this article, we investigate the computational offloading problem in a system of EH-powered devices performing correlated AIoT tasks. For each device, we make an individual decision, such as whether a device should perform the AIoT task by itself, transmit captured data for processing on another more computationally resource-rich edge device, or, due to correlations, preserve energy and allow a nearby device to perform the task. For such a problem, we base our solution on multiagent reinforcement learning (MARL) to make decisions for each device. In addition, because of the dynamic environment considered in our work, we have identified DRL as the most appropriate approach for developing an autonomous solution capable of adapting to the fluctuating energy of a device with EH in real time while taking advantage of correlation. The benefits of the learning approach in such environments have been demonstrated by many other DRL-based solutions [17], [18], [19], [20], [21], [22], [23], [24]. The contributions of this article can be summarized as follows.

1) We formulate a problem for performance optimization of EH-powered sensors performing the correlated AIoT task.
2) We propose a novel solution named balancing energy efficiency in sensor networks with MARL (BEES-MARL) that guides real-time decision-making for correlated EH-powered sensors. This solution aims to maximize performance, measured in terms of recall while being constrained by energy and bandwidth.
3) We evaluate the performance of the proposed solution in two data-driven use cases. We use established datasets [25], [26] as well as the dataset we collected for this study to simulate the environment as accurately as possible and compare our solution to two conventional task-offloading strategies and a DRL approach.
4) We demonstrate the scalability of our proposed approach to a system of N sensors.

The rest of the article is structured as follows. In Section II, we provide an extensive review of related work. In Section III, we introduce system performance metrics as well as the mathematical formulation of the proposed problem. We then describe our proposed DRL solutions in Section IV. We evaluate our proposed solutions in Section V. Finally, in Section VI, the conclusion of the work is presented.

II. Related Work

Our work is related to studies that proposed task-offloading schemes for edge devices powered by an EH [9], [10], [11], [12], [13]. For example, in [9], the mobile devices are offloading tasks to achieve a noncooperative computation offloading framework to minimize the average response time of edge devices in a stochastic environment. Min et al. [10] leveraged DRL to decide which edge device will offload the task to minimize their energy consumption and task latency. Balasubramanian et al. [11] proposed an architecture and a threshold policy to achieve energy-aware edge task offloading from EH sensors. Similarly, Liu et al. [12] relied on an online Lyapunov-based task-offloading algorithm to investigate the tradeoff between energy consumption and execution delay. Xu et al. [13] considered online learning to decide how much power a mobile edge computing (MEC), powered by an EH, allocates to each task. However, the above solutions do not optimize the performance of the data-analytics task nor aim to design a system that will ensure that EH-powered devices will avoid depleting all available energy as is the objective of our work.

Only a handful of papers examine the optimization of task-offloading schemes to improve the performance of data analytics [14], [15], [16], [27]. Lyu et al. [14] used facial recognition as an example to validate the energy effectiveness of their proposed task-offloading scheme. However, in their system, EH is not considered. Similarly, a tradeoff between object detection accuracy and latency was explored in [15] for battery-powered mobile devices. The focus of the latter was to demonstrate that task offloading can improve the frame rate and accuracy. An online learning approach was proposed in [16], which improves the accuracy of the data-analytic task while reducing the energy consumption of an edge device. In contrast, we explore a more advanced system in which devices are powered only by EH. In our approach, the system must also consider the energy that cameras will collect in the future and the correlation between cameras to achieve optimal performance.

Recently, reinforcement learning (RL) emerged as a very effective approach to resolve a plethora of problems related to the management of EH devices due to its ability to adapt to a dynamic environment [19], [20], [21], [22], [23], [24]. For example, Ait Aoudia et al. [19] proposed a RL-based power management strategy capable of maximizing the quality of service while considering available energy and energy cost of transmission. Zhang et al. [20] investigated how to employ DRL to optimize joint beamforming of reconfigurable intelligent surfaces (RISs) and base stations at the edge of the network. Chu et al. [21] employed DRL to resolve an access problem for EH devices. A balance between transmission power and modulation level to increase throughput was considered in [22]. Similarly, Sharma et al. [28] leveraged DRL to design a power control policy for EH devices to maximize throughput. Hribar et al. [23], [24] proposed the use of DRL to obtain an energy-efficient strategy for collecting correlated information. In the papers reviewed above, DRL was selected because of its adaptability to the time-varying nature of energy arrival on EH-powered devices, leading us to consider it as a suitable approach for the problem our paper is addressing as well. In our previous work [29], we proposed using a single DRL agent to make decisions on how devices should offload tasks to take advantage of correlation in a limited scenario involving two devices. In this work, we demonstrate that our newly proposed BEES-MARL performs well in large-scale scenarios with N devices, whereas a single-agent approach is suboptimal and unstable in such scenarios.

III. System Model and Problem Statement

We study a system of N sensors (e.g., cameras and LiDARs), which observe a landscape and one cloudlet performing an AIoT task. The sensors, S_n, where $n = 1, \dots, N$, transmit observations to the cloudlet for collection. The AIoT task of the system is to determine the number of objects in the observed scene, for example, vehicles crossing the intersection. Consequently, objects of interest, for example, vehicles detected by one sensor, may also be detected by another sensor. Furthermore, we assume that sensors are powered only by an EH source and have a finite energy storage. Examples of the systems under study are wireless cameras that count vehicles in an intersection, LiDAR sensors that detect the number of objects in the room, and environmental sensors that detect anomalies. An example of the system under study is shown in Fig. 1.

In the considered system, the sensors can acquire $M \in \mathbb{N}$ observations, such as LiDAR scans or images. Each sensor operates in one of the selected modes of operation, $v_n(t) \in \mathcal{V} = 1, \dots, V$. For example, one mode of operation could involve a sensor transmitting raw observations to the cloudlet, while another mode might enable sensors to process

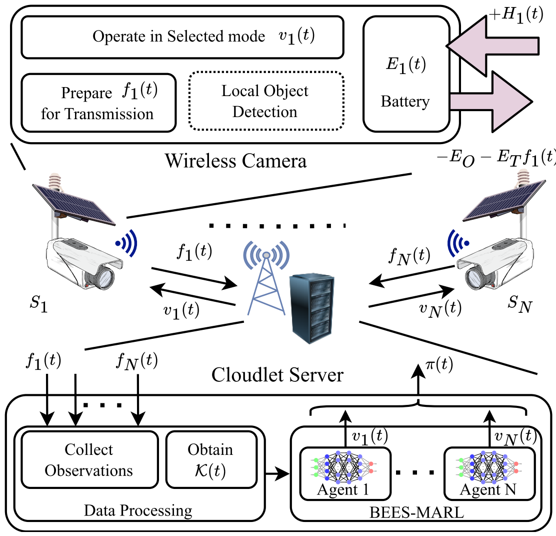


Fig. 1. Illustration of the system model featuring multiple wireless cameras and key building blocks of the proposed BEES-MARL.

observations using on-board resources to conserve energy. An example of the latter is a camera equipped with dedicated hardware capable of efficient image processing, such as Edge TPU, which can perform local object detection or transmit raw observations to the cloudlet for processing. Consequently, in each time slot $t \in \mathcal{T} = 1, 2, \dots, T$, the cloudlet must decide on the operation policy $\pi(t) = \{v_1(t), v_2(t), \dots, v_N(t)\}$. This decision directly impacts recall, latency, and energy consumption within the system.

A. Correlation in Processed Observations

We define a set $\mathcal{C}_n(t)$ as the set of all objects that the nth sensor can detect in time step t. It is important to note that, due to practical limitations of the considered system, such as objects being hidden behind one another, the observation collected by sensor n may not contain traces of every object in the observed scene. Consequently, we can define the set of all detectable objects as

$$\mathcal{C}(t) = \mathcal{C}_1(t) \cup \mathcal{C}_2(t) \cup \dots \cup \mathcal{C}_N(t) \tag{1}$$

which represents the ground truth, that is, the set of all objects that the system can detect. In our work, we only consider an object to be correctly detected if we can correctly classify it into one of the predefined categories and the position of the object can be delimited with a bounding box whose dimensions and position accurately reflect the spatial extent and position of the object in the real world. In addition, due to correlations in the processed observations, it is possible to detect the same objects in observations obtained by multiple sensors. For example, the correlation in the processed observation from the ith and jth sensors can be defined as the intersection of the two sets of detectable objects, denoted as $\mathcal{C}_j(t) \cap \mathcal{C}_i(t)$.

At time step t, $\mathcal{K}_n(t|v_n(t))$ represents the objects detected by the nth sensor based on its mode $v_n(t)$. This set is a subset of all potentially detectable objects $\mathcal{C}_n(t)$, that is, $\mathcal{K}_n(t|v_n(t)) \subseteq \mathcal{C}_n(t)$. Ideally, $\mathcal{K}_n(t|vn(t))$ is equal to $\mathcal{C}_n(t)$

if the sensor has detected all objects. If the sensor does not transmit an observation or perform an AIoT task, the set is empty, that is, $\mathcal{K}_n(t|v_n(t)) = \emptyset$. The complete set of objects that the system detects in the time step t is the union $\mathcal{K}(t|\pi(t)) = \mathcal{K}_1(t|v1(t)) \cup \mathcal{K}_2(t|v2(t)) \cup \cdots \cup \mathcal{K}_N(t|v_N(t))$.

B. Measuring Performance

To measure the performance of the system, we identified recall as the most appropriate metric. Recall measures the ability of the system to identify all relevant objects within the observed scene. This metric ensures high detection rates and is essential in scenarios where a missed object detection has significant consequences, for example, safety monitoring [30]. Consequently, in such a scenario as considered in our work, recall provides an advantage over other commonly used metrics such as precision, which quantifies the accuracy of positive predictions among all positive reports. Recall provides an advantage over precision because the consequences of failing to identify positive cases are more severe and costly as they are potential false positives. We refer to recall as $\phi(t)$, and it is defined as follows:

$$\phi(t) = \frac{|\mathcal{K}(t|\pi(t))|}{|\mathcal{C}(t)|}. \tag{2}$$

Note that recall is limited to an interval $\phi(t) \in [0, 1]$. The higher the recall is, the better the performance. However, to properly evaluate the studied system, other system parameters such as latency also need to be considered.

C. Latency

We assume that the system can adopt various wireless technologies, such as cellular, Wi-Fi, and others, to transmit observations. In the considered system, the amount of data transmitted $f_n(t|v_n(t))$ by the nth sensor depends on the selected mode of operation. For example, if the camera is set to transmit all captured images, the amount of data transmitted is proportional to the image size, that is, $f_n(t|v_n(t)) = Mf_{\text{IM}}$, with f_{IM} denoting individual image sizes in bits. In contrast, if the sensor decides not to transmit anything, the amount of data transmitted is zero, that is, $f_n(t|v_n(t)) = 0$. In addition, the delay also depends on the time required to preprocess the data on the sensor and the time required by the cloudlet to process the transmitted information. We denote the former by $\tau_D(t|v_n(t))$ and the latter by $\tau_C(t|v_n(t))$. Note that these two delays also depend on the selected operating mode. For example, if the sensors choose to process observations, the resulting τ_D is longer than when transmitting raw data. The opposite is true for τ_C. Then, we define the system latency $\mathcal{L}(t)$ as

$$\mathcal{L}(t) = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{(f_n(t|v_n(t)))}{\mathcal{B}(t)} + \tau_D(t|v_n(t)) + \tau_C(t|v_n(t))\right) \tag{3}$$

where $\mathcal{B}(t)$ represents the data rate. Unfortunately, the data rate is finite. Therefore, if the data demand from any individual sensor exceeds the available data rate, that is, $f_i(t) \geq B(t), \forall i \in 1, 2, \ldots, N$, it will result in loss of data, leading to degraded performance. Furthermore, latency, as defined in our work, could be understood as service time, encompassing the time required to transmit and process the collected data. Next, we define down-time, an energy-related system performance indicator revealing how well a system manages available energy.

D. Energy Model

In each time step, the nth sensor receives harvested energy $H_n(t)$ proportional to the current and the voltage from the EH unit. We assume that the devices are powered using a photovoltaics (PVs) as such EH schemes can reliably provide the necessary power for considered sensors. To store the harvested energy, each sensor is equipped with a battery with a maximal capacity E_M. Consequently, the sensor's available energy $E_n(t)$ is limited to an interval $E_n(t) \in [0, E_M]$. Furthermore, we assume that in each time step, sensors consume energy to operate, which we denote as E_O. This energy consumption represents the operational energy that a sensor requires to support its basic operation, such as capturing data and powering on-board elements. Finally, we define the nth sensor's available energy as

$$E_n(t) = E_n(t-1) + H_n(t) - E_O - E_T f_n(t|v_n(t)) \tag{4}$$

where E_T is the energy the system requires to transmit a bit of information.

Often, an individual sensor can deplete all its available energy, meaning that $E_n(t) = 0$. In such cases, the sensor is unable to transmit observations. However, to the system, the problem becomes severe only when none of the sensors can transmit information. To measure such instances, we introduce the concept of downtime, denoted as T_D, which specifically tracks instances when none of the sensors in the system possesses the required energy to transmit data packets.

E. Problem Statement

The main goal of the system under study is to find a policy $\pi(t)$, that is, modes of operation for every sensor, at each time step t, that will maximize the average recall. First, we define the average recall as

$$\overline{\phi} = \frac{1}{T}\sum_{t=1}^{T}\frac{|\mathcal{K}(t)|}{|\mathcal{C}(t)|}. \tag{5}$$

To achieve the set objective, the system must select a policy $\pi(t)$ in such a way as to take advantage of the correlation in the processed observations, as defined in (1). In addition, the system must also overcome other limitations of the environment, such as the limited energy that sensors collect using the EH and the limited bandwidth to maximize recall. We formulate the problem as follows:

$$\begin{aligned} &\max_{\pi(t)\in\Pi} \ \overline{\phi} \\ &\text{s.t.} \ \sum_{n=1}^{N} f_n(t, |v_n(t)) \leq \mathcal{B}(t) \\ &\qquad E_n(t) \geq 0 \quad \forall n = 1, \ldots, N \\ &\qquad t \in \mathcal{T} = \{1, 2, \ldots, T\}. \end{aligned} \tag{6}$$

To find the policy $\pi(t)$ that maximizes average recall in an environment with limited energy and bandwidth, we turn to DRL. The goal is to find a policy that allows the system to select and process observations from the smallest set of sensors while ensuring the detection of every object in the set $\mathcal{C}(t)$. Accomplishing this objective requires leveraging correlations present in the processed observations. However, considering the complexity and the vast number of possible actions, which amounts to V^N, it is crucial to employ an approach capable of exploration and learning correlations. This will enable the system to discover a well-defined and effective policy.

IV. Proposed BEES-MARL Solution

The system must find a policy that selects the operating mode of sensors that maximizes the average recall in an energy-constrained and time-varying environment. Finding such a policy is a nontrivial task, especially considering that the correlation in the processed observations is unknown to the system. Fortunately, due to the nature of the system under consideration, it can be learned over short periods of time, for example, when the same vehicle crosses the intersection for a few time steps. To achieve this, we propose a BEES-MARL solution that can optimize system performance in such an environment due to its adaptability, assuming the following conditions.

1) *Correlation:* In processed observations where sensors detect overlapping objects, accurate detection can be achieved using only a subset of the available sensors.
2) *Energy Consideration:* Sensors have limited energy available or collect energy with EH, requiring efficient energy management.
3) *Limited Bandwidth:* Bandwidth is restricted, meaning excessive data transmission by sensors may be detrimental to the performance.
4) *Computation Availability:* Cloudlet computing resources are available to support processing tasks near the network edge.

The solution we propose is based on the widely recognized framework of MARL [31], where individual learners, that is, agents, with a common signal, that is, shared reward, learn to cooperate to achieve the common goal. The chosen approach fits very well as we have a highly cooperative environment with a large number of possible combinations (V^N in total), which significantly affects the effectiveness of alternative single-agent solutions.

BEES-MARL consists of N agents, and to be able to adopt RL-based approach, we define state space $\mathcal{S}$, list available action space $\mathcal{V}$, and describe the reward signal $r(t)$.

State: The state $\mathbf{s}(t) \in \mathcal{S}$ comprises the residual energy in each device, the recall of the object detection task, and the number of detected objects, as follows:

$$\mathbf{s}(t) = \left\{\phi_M(t), |\mathcal{K}(t)|, \frac{E_1(t)}{E_M}, \ldots, \frac{E_N(t)}{E_M}\right\} \tag{7}$$

with $\phi_M(t)$ representing the recall that the system obtains at the end of each time step. The selected state space is intentionally minimalistic but contains all the relevant information the agents need to make an informed decision. The energy levels of sensors provide the agent with information about the amount of available energy in the system and as such can also serve as a hint of actions other agents are more likely to take. For example, if an agent knows that it has more energy available than others, it will more often choose a mode of operation that consumes more energy. This results in better performance, by compensating for other agents choosing a less energy-intensive mode of operation without reducing overall performance. Moreover, even if the granularity of the discretization of the inputs is low, the number of states increases extremely rapidly to thousands. For example, even using only 100 states for the energy level and a relatively low granularity of 0.025 for the recall state, the number of possible states is in the millions for $N = 4$. This is a rationale for designing BEES-MARL based on a neural network-based deep approach rather than tabular RL.

Actions: Each agent in the proposed BEES-MARL solution selects the mode of operation for the sensor to which it is assigned. For example, agent 2 decides what the mode of operation should be for sensor 2. In total, each agent has V available operation modes, that is, the available actions for any sensor are $\mathcal{V} \in \{1, \ldots, V\}$. The decision is made at each decision epoch, which occurs at the start of the time step for the next time step, based on the state $s(t)$. The agent selects the action based on the Q-values for the given state, with Q representing the learned long-term quality of the action. The higher the Q value for a given action, the more desirable that action is. At each decision point, the agent evaluates the Q-values for all possible actions in the current state using the neural network. With a probability ϵ, the nth agent chooses a random action to explore new possibilities. Otherwise, with probability $1-\epsilon_n$, it selects the action $v_n(t)$ that maximizes the Q-value

$$v_n(t) = \arg\max_{v_n(t)\in\mathcal{V}} Q(s(t), v_n(t); \theta_n) \tag{8}$$

where θ_n represents the parameters of the neural network. Over time, ϵ_n is typically decayed, allowing the agent to focus more on exploiting the learned policy as it gains more experience. This policy selection enables our multiagent solution to effectively tackle complex scenarios by leveraging the distributed intelligence of the agents learning from the selected state space and, as we demonstrate in Section V, leads to a robust and adaptive solution.

To determine $\phi_M(t)$ in our solution, we are faced with the challenge of the ground truth being unknown to the system. To overcome this challenge, the system implements a periodic mechanism wherein each sensor transmits an observation captured at the same moment. This principle is exemplified by the camera sensor in Fig. 2. By following such an approach, the system obtains a set of every object in the observed scenery as close as possible to the ground truth, that is, $\mathcal{C}$. In practical terms, each sensor in the system transmits the Mth raw measurement in every time step. This approach is necessary because only when every sensor in the system transmits observations can the system get as close to the ground truth as possible. The calculation of $\phi_M(t)$ is defined

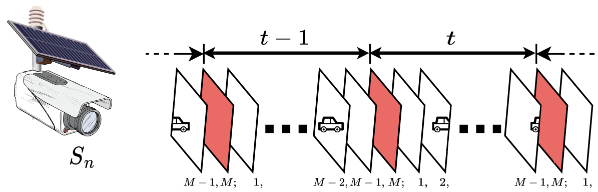


Fig. 2. At the end of each time step, every sensor in the system transmits the Mth raw measurement to the cloudlet server, while other observations $1, \ldots, M-2, M-1$ are transmitted according to the selected policy.

as follows:

$$\phi_M(t) = \frac{|\mathcal{K}(t|_{M-1})|}{|\mathcal{K}(t|_M)|}. \tag{9}$$

In our preliminary studies involving the analysis of the video dataset [25], we observed that employing the aforementioned approach results in ϕ_M being the same as the recall determined using (2) in 99.9% of cases. This indicates that the system accurately calculates recall as it would if it had access to the ground truth.

Reward: In our system, the reward signal is derived directly from the primary objective of maximizing recall and is defined as follows:

$$r(t) = (\phi_M(t) - 0.5) \times 2. \tag{10}$$

We have intentionally kept the reward signal simple to conform to the prevailing view within the DRL community. This view holds that DRL agents can learn complex behaviors to improve their performance, even in the face of environmental complexity, as noted by Silver et al. [32]. The proposed reward signal is crucial for promoting cooperation between agents, as it requires the development of strategies to maximize recall. To maximize recall, agents need to adopt a collaborative approach to decision-making, that is, action selection. In the highly collaborative environment we consider, the simultaneous transmission of raw observations by each agent may lead to inefficiencies due to bandwidth limitations and energy depletion. Therefore, by concentrating on maximizing recall, the system also reduces energy and bandwidth consumption, as it leads to fewer tasks or transmissions being performed by sensors overall. Moreover, the reward signal encompasses both positive and negative values, allowing for faster convergence toward improved policies. Such an approach to reward design is applicable in the context of using DRL for sensor networks because it effectively quantifies the performance of the system relative to the desired goal. As we will show in the following section through experiments, despite the simplicity of the reward signal, the agents can learn complex behaviors that maximize recall.

We outline the proposed BEES-MARL solution in Algorithm 1. At the start, the proposed algorithm initializes N policy and target networks along with N replay memories (line 1). The algorithm proceeds with the observation of the initial state (line 2). The action selection, that is, the selection of the policy $\pi(t)$, is carried out in parallel for N sensors. Action selection is performed using the epsilon-greedy strategy (line 4 and 5), which establishes a balance between exploration and exploitation by randomly selecting actions with probability ϵ_n (exploration) and the best-known action with probability $1-\epsilon_n$ (exploitation). After the action has been selected, it is transmitted to the sensor (line 6). The system then waits for the sensors to operate in the selected mode and waits until the Mth raw measurements are received from each sensor (line 7). The algorithm then observes a new state (line 9) and stores the new experience in the replay memory (line 10). In the next step, the algorithm trains neural networks by first randomly selecting J experiences from the replay memory (line 11). The target values are then determined for each state transition (line 13). The target values are calculated by adding the reward to the discounted maximum Q-value for the next state as predicted by the target network. With the determined target values, the algorithm performs the gradient descent, where the parameters of the network are updated with the loss function (line 15). In the last training step, the weights of the target neural network are softy updated with a factor τ (line 17). Note that, in practice, steps 11–17 do not have to be executed sequentially. For example, while waiting to receive raw measurements from sensors (line 7). This means that the system can go from step 10 directly to step 6 to ensure real-time system response.

Algorithm 1 Proposed BEES-MARL Solution

1: Randomly initialize N policy networks $Q_n(\mathbf{s}_n, v_n|\theta_n)$, initialize N target networks Q'_n with weights $\theta'_n \leftarrow \theta_n$, and N replay memories $\mathcal{D}_n$ to capacity D, with $n \in \{1, 2, \ldots, N\}$
2: Observe initial state $\mathbf{s}_n \forall n \in \{1, 2, \ldots, N\}$ at $t = 0$
3: **for** $t = 1, T$ **do** $\triangleright$ In parallel $\forall n \in \{1, 2, \ldots, N\}$
4: With probability $1-\epsilon_n$ select action $v_n(t)$ using Eq. 8
5: Otherwise select action $v_n(t)$ at random
6: Set n-th sensor operating mode as defined by $v_n(t)$
7: $\triangleright$ Wait until M-th raw measurement is received.
8: Determine ϕ_M using Eq.(9) and $r(t)$ with Eq.(10)
9: Observe new state $\mathbf{s}_n(t)$
10: Store experience $\mathbf{s}_n(t\text{-}1), \mathbf{s}_n(t), r(t), v_n(t)$ in $\mathcal{D}_n$
11: Sample random batch of J experiences from $\mathcal{D}_n$
12: **for** every $\{\mathbf{s}_n(j\text{-}1), \mathbf{s}_n(j), r(j), v_n(j)\}$ in batch **do**
13: Set $y_n(j) = r(j) + \gamma max_{v_n(j+1)} Q'(\mathbf{s}_n(j\text{+}1), v_n(j\text{+}1))$
14: **end for**
15: Calculate loss: $\mathcal{Z}_n = \frac{1}{J}\sum_{j=0}^{J-1}(Q_n(\mathbf{s}_n(j), v_n(j)) - y_n(j))^2$
16: Update $Q_n(\mathbf{s}_n, v_n|\theta_n)$ by minimizing the loss $\mathcal{Z}_n$
17: Softly update the n-th target network: $\theta'_n \leftarrow \lambda\theta_n\text{+}(1\text{-}\lambda)\theta'_n$
18: **end for**

We implemented BEES-MARL using the Pytorch Python library [33]. The hyperparameters of the proposed solution, listed in Table I, were selected through grid search to fine-tune the performance of the agents during training. The neural network of each agent consists of two hidden layers, as depicted in Fig. 3, with the first hidden layer configured with 32 neurons and the second with 64 neurons. Various network configurations were experimented with; however, smaller networks resulted in reduced performance, while larger

TABLE I
BEES-MARL HYPERPARAMETERS

Hyperparameter	Value	Hyperparameter	Value
Batch size J	1024	Memory size D	$2 * 10^5$
Optimizer	Adam	Start epsilon value ϵ	0.99
Loss Function	MSE	Epsilon decay	0.9998
Target ANN soft update τ	10^{-3}	Epsilon min value ϵ	0.2
Number of layers L	2	Number of neurons W_1, W_2	32, 64

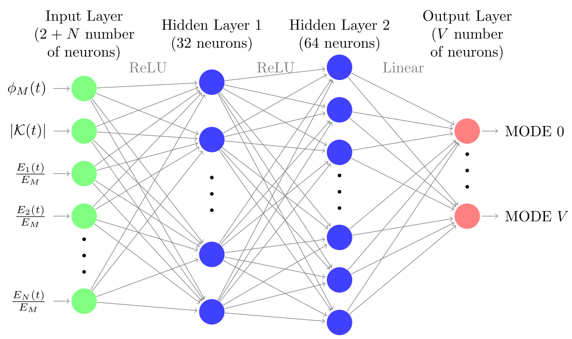


Fig. 3. Neural network structure of the nth agent.

networks with more neurons or hidden layers did not yield any performance improvement. The ReLU activation function was applied to both hidden layers due to its computational efficiency, as it involves only a simple thresholding operation compared to other standard activation functions such as sigmoid or tanh. The output layer was activated with a simple linear function, where each output neuron represents a Q-value corresponding to an action. For example, the second neuron is associated with the second mode of operation; thus, when the second neuron has the highest value for a given state, the agent should select the sensor to operate in the second mode.

The size of the neural network, together with some other factors such as the dimension of the action space and the dimension of the state space ($|\mathbf{s}|$), determines the computational complexity of our proposed solution. We can express it as follows:

$$\mathcal{O}\left(NV|\mathbf{s}(n)|\prod_{l=1}^{L} W_l\right) \tag{11}$$

where L is the number of layers and W_l is the number of neurons in the lth layer. Note that due to the relatively small size of the neural network, the overall complexity is relatively low and scales linearly with the number of sensors in the system. In Section V, we demonstrate that the proposed solution has significant advantages in terms of performance and stability for larger systems compared to heuristic approaches and other conventional DRL-based solutions.

BEES-MARL is entirely executed in the cloudlet, and only decisions, that is, selected mode of operation, are transferred to the sensors. However, the decisions are made in a decentralized manner, that is, individually for each sensor. We have chosen a cloudlet as the location for the implementation of both proposed solutions as it provides the necessary computational power for training the DRL agent and offloading energy-intensive tasks to resource-rich devices. By conducting all computations within the cloudlet, a comprehensive and synchronized approach can be achieved to optimize system performance, address intricate decision tasks, facilitate efficient communication, enable centralized learning, and simplify implementation and management processes. In short, BEES-MARL makes decisions individually for each sensor to facilitate scalability.

V. DATA-DRIVEN EVALUATION

In this section, we first evaluate the proposed BEES-MARL in two purely data-driven use cases, to show the benefits of taking advantage of correlation in processed observations, and, then also in a general setting, to demonstrate the scalability of our proposed approach. The first use case is based on a system with two cameras counting vehicles at an intersection from the Ko-PER dataset [25]. The second use case is a system with four LiDAR sensors that count the number of people in a room, which we collected in May 2022 at Shibarura Institute of Technology, Japan. In the general setting, we use an exponential function to represent the correlation and test the performance in a system with up to 20 sensors performing correlated data-analytic tasks. We simulate the performance over ten days, but the reported results are based on the average of the last seven days. Furthermore, we conducted each experiment five times and reported the average value obtained.

We compare the performance of the proposed approach with greedy and energy-aware strategies, along with a centralized single-agent DRL-based solution. The sensors using the greedy strategy always choose to transmit, provided they have sufficient energy to perform the transmission. The energy-aware strategy is based on heuristic energy-efficient task scheduling algorithms [34]. In the energy-aware approach, the system selects two-thirds of the sensors with the most energy available to transmit or process information. The DRL-based solution is repurposed from deep Q-network (DQN) [35], a well-recognized and widely adopted DRL method implemented as described in our prior work [29].

In our simulation, we utilize a PVs testbed located at the University of Castilla-La Mancha in Ciudad Real (Spain) [26] to determine the arrival of energy for the PVs EH scheme, enabling us to model $H_n(t)$ values directly based on real-world energy collection. We use measurements obtained between 16th and 30th of August 2018 as input to our simulation, which allows us to test the performance of the proposed system over 15 days. The nominal power of the solar panels used is 2 W, and, in practice, it is expected that multiple panels will be used to power a wireless camera. We assume that each of the cameras is equipped with four panels with 80% efficiency, that is, $\eta_{\text{eh}} = 0.8$. We list the remaining simulation parameters in Table II which we modeled based on existing measurements. For example, we model $E_O(t)$ according to the consumption of a Raspberry Pi equipped with a sensor and the sensor consumes the said amount of energy regardless of how it operates.

A. Cameras Observing Intersection

The first use case is based on the intersection dataset Ko-PER [25] and focuses on a system with two cameras.

TABLE II
SIMULATION PARAMETERS

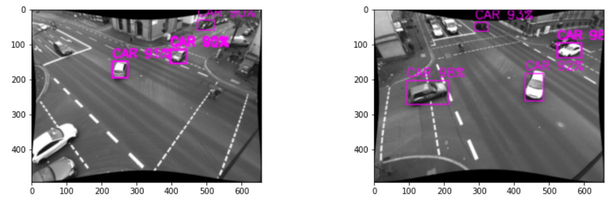
Parameter	Value	Parameter	Value
M	10	time step size	$1s$
η_{eh}	0.8 E_M	$185kJ$	
E_O	$139mJ$		

Ko-PER consists of monochrome camera images and raw laser scanner measurements for one intersection. The data sequence is six and a half minutes long, or 9670 frames obtained from two different viewpoints, that is, two different cameras. Therefore, in this analysis, we had to limit the number of sensors to two, that is, $N = 2$. We also extrapolate the data by resampling. The images from Ko-PER are captured at 25 frames/s. However, in our simulation, we set M to ten, which is the standard for traffic cameras [36]. To avoid correlation between samples, we randomly change the starting point every minute in the sequence from which our simulation obtains images. Each sensor has three different modes of operation: 1) transmitting raw image to the cloudlet for processing; 2) using the local object detector; and 3) transitioning to the standby mode, that is, $V = 3$. This means that the proposed BEES-MARL approach has to choose between three modes of operation for every sensor.

We use you only look once (YOLO) version 3 [37] for object detection on the cloudlet and tiny-YOLO version 4 for object detection on the camera. For local object detection, that is, using tiny-YOLO, we assume that the sensor consumes 57.5 mJ of energy on average. However, the amount of data transmitted will be less since the sensor will only report the number of objects detected. We assume that a device can process the captured images with dedicated hardware capable of processing images efficiently. Edge TPU, NVIDIA Xavier, and NovuTensor are examples of such specialized hardware units that can ensure that the energy cost for processing on the edge device is lower when the energy cost for transmitting the captured images [38] is high. For example, to transmit the image, the sensors must transmit an average of 162 KB of data, but transmitting the processed information requires only 1.5 KB of data. Note that the energy-aware approach in this use case selects the transmission of images from the camera with more energy, while the camera with less energy processes the information.

Fig. 4 shows an example of how our system detects and counts vehicles. When processing images on cloudlet, the system detects two vehicles from camera 1 and three from camera 2. Note that only the white vehicle at the bottom left, as seen from camera 1 [see Fig. 4(a)], has entered the intersection. The same is true for the vehicles at the top of the image from camera 2 [see Fig. 4(b)]; these vehicles have already left the intersection or have not yet entered it. In this case, the system can process only the information from camera 2 and still get correct information, that is, $\phi = 1$.

In Fig. 5, we show the recall, latency, and downtime as we vary the energy cost of transmission while keeping the bandwidth constant. The recall and latency decrease with the increase of energy cost as expected for all four approaches.



(a) (b)

Fig. 4. Example of images obtained from the Ko-PER dataset at the same time step from both cameras and objects detected by YOLO. (a) Camera 1. (b) Camera 2.

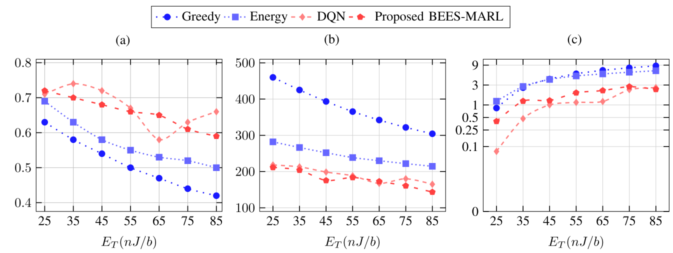


Fig. 5. Recall, latency, and downtime depending on the energy cost of transmission for different approaches, $\mathcal{B} = 3$ MBps. (a) $\overline{\phi}$. (b) $\mathcal{L}$(ms). (c) T_D(h/day).

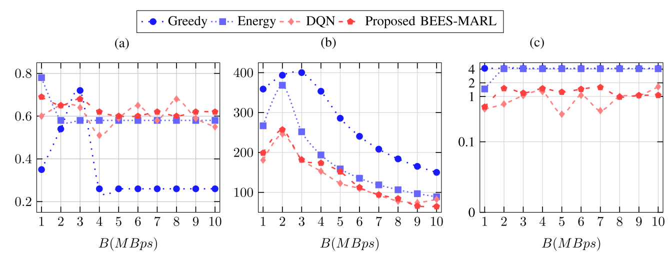


Fig. 6. Recall, latency, and downtime over bandwidth for different approaches, $E_T = 45$ nJ/b. (a) $\overline{\phi}$. (b) $\mathcal{L}$(ms). (c) T_D(h/day).

The downtime, on the other hand, increases as the system requires more energy to transmit data. The greedy approach does not perform well due to bandwidth limitations, preventing the system from achieving high recall when the energy cost of transmitting is low. The energy approach performs much better, but both DQN and our proposed solution result in much better overall performance. The performance between DQN and the proposed BEES-MARL is comparable. However, we can observe a fluctuation in the DQN approach performance, with the most noticeable occurrence observed when the energy transmission cost is 65 nJ. In such a case, the DQN learns a policy that achieves better recall but results in higher latency. In addition, the more energy-constrained the system is, the better the performance of our proposed BEES-MARL solution is when compared to the DQN approach.

We show similar performance in Fig. 6 where we change the bandwidth while keeping the energy cost of transmission constant. Interestingly, the system's performance peaks for all four approaches when the bandwidth is around 2–3 MBps. The peak occurs because the system bandwidth is insufficient to transmit all M images from both sensors at each time step, forcing the system not to send some images, which in turn affects recall. When the bandwidth is 4 MBps or more, the system can easily transmit data from both sensors

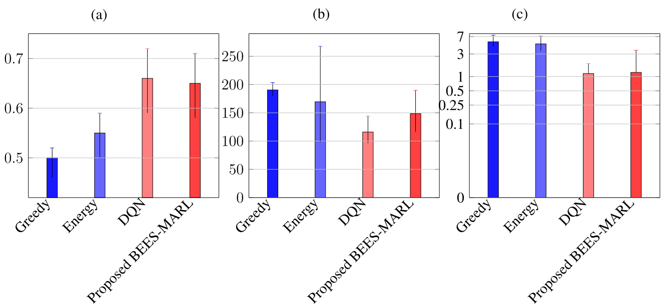


Fig. 7. Recall, latency, and downtime in a dynamic environment for a system of cameras observing intersection. (a) Daily $\overline{\phi}$. (b) $\mathcal{L}$(ms). (c) T_D(h/day).

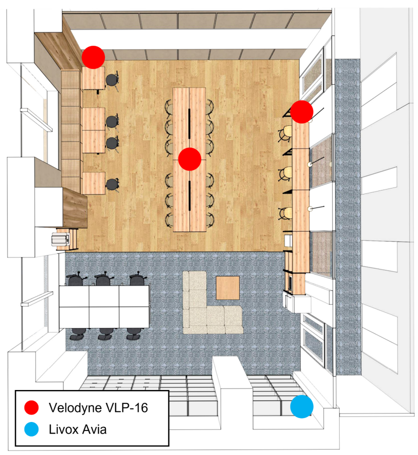


Fig. 8. Arrangement of the LiDAR sensors in the laboratory.

if necessary. Therefore, recall and downtime are largely unaffected when the bandwidth is higher. However, latency decreases because the system can transmit information faster. Similar to the previous findings, the DQN approach exhibits a fluctuating downward spiral, whereas our proposed approach BEES-MARL demonstrates a stable trend.

Fig. 7 shows the achieved daily values of recall, latency, and downtime for each policy when the energy cost of transmissions and bandwidth vary randomly. The DQN and proposed solution achieve significantly better performance compared to the greedy and energy-aware strategies, showing that learning to adapt to changes in the environment improves performance. Moreover, the DRL-based solutions can achieve better recall with lower latency and downtime. While the DQN exhibits marginally better results in terms of recall, latency, and downtime, our observations from a static environment suggest that the proposed BEES-MARL solution offers greater stability, making it the preferred solution for the given system.

B. System of LiDAR Sensors

The dataset for this case study is based on measurements we collected in May 2022 in the Ryoichi Shinkuma Laboratory at Shibaura Institute of Technology, Japan. The dataset captured the motion of five students in the laboratory with over 10 000 datapoints and over three and a half minutes in length. We used four LiDAR sensors, three Velodyne VLP-16 [39], and one Livox Avia [40]. We arranged them in the laboratory as shown in Fig. 8, with the Livox Avia sensor placed in one corner and

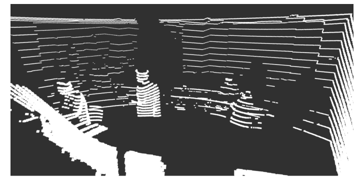


Fig. 9. Visualization of the point cloud from one LiDAR sensor.

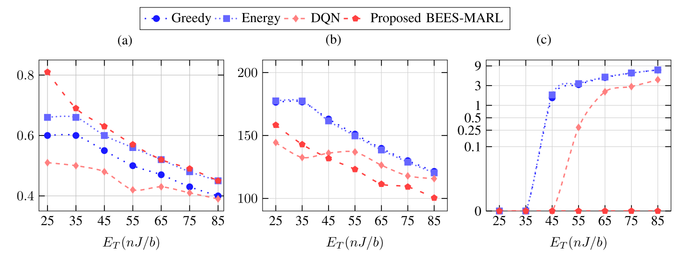


Fig. 10. Recall, latency, and downtime depending on the energy cost of transmission for different approaches, $\mathcal{B} = 10$ MBps. (a) $\overline{\phi}$. (b) $\mathcal{L}$(ms). (c) T_D(h/day).

two Velodyne LiDARs in opposite corners, while the third Velodyne LiDAR was positioned in the center of a table in the middle of the laboratory. The locations were chosen to maximize the coverage of each sensor. Sensors are calibrated to the environment, and to detect a moving object, for example, a person, the system analyzes the changes in the integrated point cloud, as visualized in Fig. 9. An object is detected only if a sufficient number of new points are detected; in our experiment, we set the sensitivity threshold to 200 new points.

In the considered indoor environment, EH techniques such as radio frequency (RF) harvesting [41], offer potential for powering the sensors. In addition, thermal and vibration EH methods could be employed. In our validation, we iterate over the LiDAR dataset (as we have iterated over the Ko-PER dataset in the study presented in Section V-A). Each sensor has two different modes of operation: 1) transmitting the raw image to the cloudlet and 2) transitioning to the standby mode, that is, $V = 2$. Consequently, the proposed BEES-MARL solution has to select between two actions for each sensor. Sensors captured ten frames per second, that is, $M = 10$, with each frame data size of 480 KB on average for the Velodyne sensor, and 384 KB for Livox Avia. As a result, the energy cost of transmitting observations is higher due to the significantly larger amount of data compared to the previous use case. However, the operating energy (E_O) remains at the same level as in the previous use case.

First, in Fig. 10, we show the system performance when we increase the energy cost of transmission. In terms of recall, the BEES-MARL approach gives the best performance, while the performance of DQN is lower than the recall that can be achieved with greedy and energy-aware strategies. In terms of latency and downtime, the DRL solutions outperform the two heuristic approaches, with BEES-MARL even achieving zero downtime. Interestingly, as the energy cost of the transmissions increases, the proposed BEES-MARL approach is comparable with the energy-aware strategy, but it achieves much better performance in terms of latency and downtime. Such a result is due to the ability of the BEES-MARL approach to better manage limited resources. Similar to the previous use

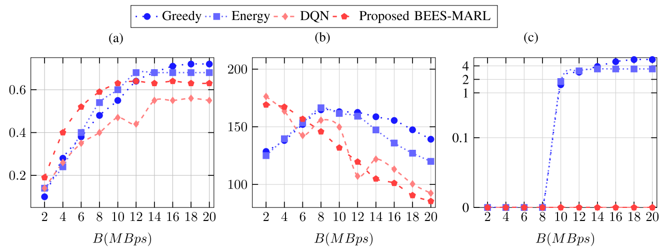


Fig. 11. Recall, latency, and downtime over bandwidth for different approaches, $E_T = 45$ nJ/b. (a) $\overline{\phi}$. (b) $\mathcal{L}$(ms). (c) T_D(h/day).

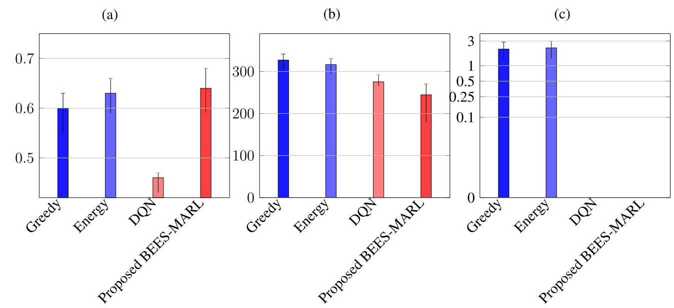


Fig. 12. Recall, latency, and downtime in a dynamic environment for a system of LiDAR sensors. (a) Daily $\overline{\phi}$. (b) $\mathcal{L}$(ms). (c) T_D(h/day).

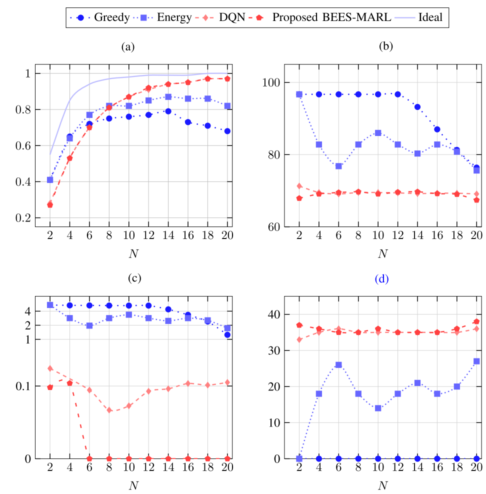


Fig. 13. Recall, latency, and downtime in the static environment depending on the number of sensors N in the system, $E_T = 45$ nJ/b and $\mathcal{B} = 50$ MB. (a) $\overline{\phi}$. (b) $\mathcal{L}$(ms). (c) T_D(h/day). (d) Reduced data transmission(%).

case, the DQN approach displays fluctuations and unstable performance.

Second, we show the performance of the system as the bandwidth increases in Fig. 11. At low bandwidth, the proposed BEES-MARL achieves better recall compared to other approaches. However, as the bandwidth increases and the system becomes less constrained, the greedy and energy-aware strategies become more desirable in terms of recall. For latency, the reverse is true: for the DQN and proposed solutions, latency decreases, while for the greedy and energy-aware strategies, it first increases and then decreases. As for downtime, DQN and BEES-MARL achieve zero downtime, while the energy strategy leads to about 2 h of downtime per day and the greedy strategy leads to about 5 h/day. These results provide insights into how delay affects performance. As the bandwidth increases, recall and downtime stabilize, while latency continues to decrease. This indicates that higher bandwidth, once it ceases to be a bottleneck for the system, will only result in lower latency and not improved recall and downtime.

The system performance under random variation of E_T and $\mathcal{B}$ is shown in Fig. 12. Overall, the proposed BEES-MARL achieves the best performance. The DQN approach results in no downtime and the second-best latency performance. Even with a relatively small number of actions, the BEES-MARL approach outperforms the DQN approach. We also discovered an interesting feature of the system performing a correlated data-analytic task. Sensors that can detect more objects than others are more often selected for transmission by our proposed BEES-MARL solution. A discovery we made when trying to understand why some of the sensors consume energy faster than others. Not every sensor performs its task equally well. A more detailed analysis of action selection in the use case of correlated cameras and LiDAR sensors shows that some sensors detect more objects than others. Consequently, the system tended to consume energy from camera 1 much faster. In the case of the LiDAR use case, one of the LiDAR sensors was much better at detecting students in the room, and our proposed solution can take advantage of this.

C. System of N Sensors

In the last part of our evaluation, we test the performance of the system when the number of sensors increases.[1] To model the correlation,[2] we identified the following function:

$$f(N) = x \exp(-yN) + z \tag{12}$$

as the most appropriate model correlation in the data-analytic task, where the parameters $x = -7.59$, $y = 0.58$, and $z = 5.09$ are extracted from the LiDAR dataset. We use the average correlation for each combination of four sensors to extract the parameters. Such a model allows for a flexible representation of the data, where changes in one variable lead to exponentially increasing or decreasing changes in another variable. Furthermore, by extracting the parameters directly from the LiDAR dataset, the model represents how the correlation in the processed information evolves with the number of sensors in the system. The sensors transmit up to ten images per second, that is, $M = 10$, and the average frame size is 384 KB. We also assume that the sensors have two modes of operation: Transmitting the raw image to the cloudlet or transitioning to the standby mode, that is, $V = 2$.

Fig. 13 shows the performance of the system over the number of sensors when we keep the transmission energy cost and bandwidth constant. When the number of sensors is low, the greedy and energy-aware strategies achieve higher recall

[1]Implementation code is available at github.com/hribarjernej89/BEES-MARL

[2]The exponential function was selected because it provided the best correlation fit in the processed LiDAR dataset, outperforming models like logarithmic and quadratic.

TABLE III
QUALITATIVE ANALYSIS AND COMPARISON OF THE PROPOSED BEES-MARL SOLUTION TO THE STATE-OF-THE-ART

Approach	Algorithm	Energy Considerations	Correlation	Reported Results
Online learning for MEC data analytics [16]	Heuristic	Transmission energy	No, cooperation	Accuracy gain of up to 20%
Edge Power-Modulation Balance [22]	DQN	Transmission power	No	Improved throughput of 30%
Sensor data collection scheduler [23], [24]	DDPG	Battery and EH	Yes, in observations	Extends deployment time $x2$
Distributed Power Control [28]	MARL	Limited energy (EH)	No	Throughput close to the optimal
Utilising Correlation from cameras [29]	DQN	Limited energy (EH)	Yes, task in analytics	15% better accuracy
Proposed BEES-MARL	DQN MARL	Limited energy (EH)	Yes, in task analytics	Up to 25% better recall

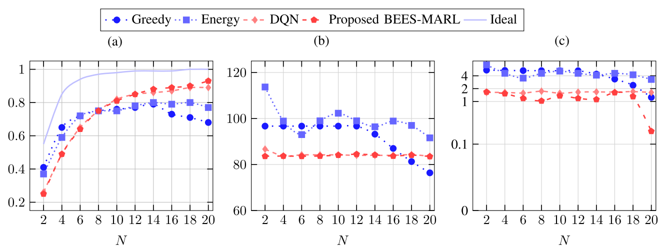


Fig. 14. Recall, latency, and downtime in the dynamic environment depending on the number of sensors. (a) $\overline{\phi}$. (b) $\mathcal{L}$(ms). (c) T_D(h/day).

than the proposed BEES-MARL solution. However, when the number of sensors in the system increases, the performance of BEES-MARL approaches the optimum, while keeping latency low and achieving zero downtime. Interestingly, DQN achieves similar performance in terms of recall and latency, with a higher downtime. Both the greedy and energy-aware strategies experience performance degradation when the number of sensors is higher, that is, $N > 16$. This degradation is due to trying to transmit more data than the bandwidth allows, resulting in information loss, which in turn affects retrieval. Overall, the proposed BEES-MARL solution achieves better performance due to its ability to adapt to changes. In addition, Fig. 13(d) illustrates the amount of data each approach preserves, that is, data that is not transmitted. Interestingly, DQN and the proposed BEES-MARL approach preserve more than a third of data transmissions with little to no impact on performance. This indicates that even when sensor information is not transmitted, the system still effectively detects objects due to the correlation. On the other hand, the greedy approach preserves no data, while the energy-aware approach preserves less than a third of the data, with a noticeable impact on performance.

The results are similar when we test performance in a dynamic environment, shown in Fig. 14, in which the cost and bandwidth vary randomly. The recall performance of BEES-MARL is up to 3% better than DQN, 16% better than the energy-aware approach, and 25% better than the greedy approach. The latency follows a similar pattern as in the static environment described above. A more noticeable difference is in the downtime, as using BEES-MARL can result in an hour or less downtime in comparison to DQN.

A closer inspection reveals a problem inherent to the DQN algorithm, as the number of available actions increases exponentially with the number of sensors. The computational complexity of the considered DQN solution is $\mathcal{O}(V^N|\mathbf{s}|W_1W_2)$ and increases exponentially with N, in contrast to the linear increase in the proposed BEES-MARL solution [expressed in (11)]. Consequently, the DQN algorithm does not fully explore the action space when the number of sensors is high, resulting in a subpar solution, a behavior already observed in the previous use case. Although the proposed BEES-MARL solution requires more computational power since the cloudlet needs to train N agents instead of one, considering the stability and improved performance, it is more desirable than other heuristic or DQN solutions.

To provide further analysis of our solution, in Table III, we qualitatively compare the proposed BEES-MARL solution to other state-of-the-art solutions that tackle similar problems or employ similar approaches. However, these other solutions address specific aspects of power control and energy management in EH sensor networks [16], [22], [23], [24], [28]. These works either do not fully exploit the potential benefits of considering correlations for task analytics [23], [24], focus only on cooperation [16], or ignore these potentials altogether [22], [28]. The exception is the work in [29], which was our initial work, but as we demonstrated in this section, it does not generalize well. Overall, the proposed BEES-MARL solution offers an approach that not only improves performance but also enhances energy efficiency, making it a more suitable solution for EH sensor networks where sensors perform specific ML or data-analytic tasks and energy preservation is critical.

VI. CONCLUSION

In this article, we explored how a sensor network can utilize correlation in processed observations to optimize energy conservation of EH-powered devices in IoT without sacrificing performance. We proposed the BEES-MARL approach to improve system performance, measured through recall while minimizing latency and preventing outages. Our results, demonstrated through two data-driven use cases and a general scalability case, show that the proposed approach can be up to 16% more accurate without incurring outages.

Our findings highlight the potential of the proposed solution for sustainable AIoT applications by enhancing the energy efficiency of EH-powered devices in many scenarios such as agriculture, smart cities, and so on. Leveraging correlation in sensor observations reduces energy consumption, contributing to longer device lifespans and reduced environmental impact. In addition, the BEES-MARL approach exemplifies the importance of data-efficient algorithms in AIoT, reducing redundant data processing and transmission to conserve energy and computational resources.

Our future work will address certain limitations of the MARL approach, such as the need for more efficient cooperation between agents and communication uncertainty. To overcome these challenges, we will explore methods like value-decomposition networks (VDNs) [42] or deep W-networks (DWNs) [43], aiming to enhance energy efficiency and overall system performance. In addition, we will investigate how our approach can contribute to the development of algorithms that dynamically adjust sensor fusion parameters based on real-time metrics and environmental conditions. A potential application of this is in automated driving within urban areas, where sensor fusion integrates data from both the vehicle and its environment [44].

Acknowledgment

The authors would like to thank Katsuki Teraoka for his help in acquiring the LiDAR measurements.

References

[1] L. Kong et al., "Edge-computing-driven Internet of Things: A survey," *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1–41, Dec. 2022.

[2] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[3] L. A. Villas, A. Boukerche, H. A. B. F. de Oliveira, R. B. de Araujo, and A. A. F. Loureiro, "A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks," *Ad Hoc Netw.*, vol. 12, pp. 69–85, Jan. 2014.

[4] J. Hribar, M. Costa, N. Kaminski, and L. A. DaSilva, "Using correlated information to extend device lifetime," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2439–2448, Apr. 2019.

[5] A. P. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, "A survey of green networking research," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 1, pp. 3–20, 1st Quart., 2012.

[6] J. Yang, Y. Han, Y. Wang, B. Jiang, Z. Lv, and H. Song, "Optimization of real-time traffic network assignment based on IoT data using DBN and clustering model in smart city," *Future Gener. Comput. Syst.*, vol. 108, pp. 976–986, Jul. 2020.

[7] J. Wen, J. Yang, T. Wang, Y. Li, and Z. Lv, "Energy-efficient task allocation for reliable parallel computation of cluster-based wireless sensor network in edge computing," *Digit. Commun. Netw.*, vol. 9, no. 2, pp. 473–482, Apr. 2023.

[8] S. Bi, C. K. Ho, and R. Zhang, "Wireless powered communication: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 117–125, Apr. 2015.

[9] T. Zhang and W. Chen, "Computation offloading in heterogeneous mobile edge computing with energy harvesting," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 1, pp. 552–565, Mar. 2021.

[10] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.

[11] V. Balasubramanian, N. Kouvelas, K. Chandra, R. V. Prasad, A. G. Voyiatzis, and W. Liu, "A unified architecture for integrating energy harvesting IoT devices with the mobile edge cloud," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Singapore, Feb. 2018, pp. 13–18.

[12] W. Liu, X. Zhou, S. Durrani, H. Mehrpouyan, and S. D. Blostein, "Energy harvesting wireless sensor networks: Delay analysis considering energy costs of sensing and transmission," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4635–4650, Jul. 2016.

[13] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 3, pp. 361–373, Sep. 2017.

[14] X. Lyu et al., "Selective offloading in mobile edge computing for the green Internet of Things," *IEEE Netw.*, vol. 32, no. 1, pp. 54–60, Jan. 2018.

[15] X. Ran, H. Chen, Z. Liu, and J. Chen, "Delivering deep learning to mobile devices via offloading," in *Proc. Workshop Virtual Reality Augmented Reality Netw.*, Angeles, CA, USA, 2017, pp. 42–47.

[16] A. Galanopoulos, G. Iosifidis, and T. Salonidis, "Optimizing data analytics in energy constrained IoT networks," in *Proc. 16th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, Shanghai, China, May 2018, pp. 1–8.

[17] J. Yang et al., "Reinforcement learning based edge computing in B5G," *Digit. Commun. Netw.*, vol. 10, no. 1, pp. 1–6, Feb. 2024.

[18] N. C. Luong et al., "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019.

[19] F. Ait Aoudia, M. Gautier, and O. Berder, "RLMan: An energy manager based on reinforcement learning for energy harvesting wireless sensor networks," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 2, pp. 408–417, Jun. 2018.

[20] T. Zhang, D. Xu, A. Tolba, K. Yu, H. Song, and S. Yu, "Reinforcement-learning-based offloading for RIS-aided cloud–edge computing in IoT networks: Modeling, analysis, and optimization," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19421–19439, Jun. 2024.

[21] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement learning-based multiaccess control and battery prediction with energy harvesting in IoT systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2009–2020, Apr. 2019.

[22] M. Li, X. Zhao, H. Liang, and F. Hu, "Deep reinforcement learning optimal transmission policy for communication systems with energy harvesting and adaptive MQAM," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5782–5793, Jun. 2019.

[23] J. Hribar, A. Marinescu, A. Chiumento, and L. A. Dasilva, "Energy-aware deep reinforcement learning scheduling for sensors correlated in time and space," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6732–6744, May 2022.

[24] J. Hribar, L. A. DaSilva, S. Zhou, Z. Jiang, and I. Dusparic, "Timely and sustainable: Utilising correlation in status updates of battery-powered and energy-harvesting sensors using deep reinforcement learning," *Comput. Commun.*, vol. 192, pp. 223–233, Aug. 2022.

[25] E. Strigel, D. Meissner, F. Seeliger, B. Wilking, and K. Dietmayer, "The Ko-PER intersection laserscanner and video dataset," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, Oct. 2014, pp. 1900–1901.

[26] M. Kuzman, X. D. T. García, S. Escolar, A. Caruso, S. Chessa, and J. C. López, "A testbed and an experimental public dataset for energy-harvested IoT solutions," in *Proc. IEEE 17th Int. Conf. Ind. Informat. (INDIN)*, vol. 1, Jul. 2019, pp. 869–876.

[27] B. Chen, Z. Yan, and K. Nahrstedt, "Context-aware optimization for bandwidth-efficient image analytics offloading," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 20, no. 9, pp. 1–22, Sep. 2024.

[28] M. K. Sharma, A. Zappone, M. Assaad, M. Debbah, and S. Vassilaras, "Distributed power control for large energy harvesting networks: A multi-agent deep reinforcement learning approach," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 4, pp. 1140–1154, Dec. 2019.

[29] J. Hribar, R. Shinkuma, G. Iosifidis, and I. Dusparic, "Analyse or transmit: Utilising correlation at the edge with deep reinforcement learning," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 1–6.

[30] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jul. 2020, pp. 237–242.

[31] A. Oroojlooy and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *Int. J. Speech Technol.*, vol. 53, no. 11, pp. 13677–13722, Oct. 2022.

[32] D. Silver, S. Singh, D. Precup, and R. S. Sutton, "Reward is enough," *Artif. Intell.*, vol. 299, Oct. 2021, Art. no. 103535.

[33] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NIPS*, Dec. 2019, pp. 8024–8035.

[34] R. Ghafari, F. H. Kabutarkhani, and N. Mansouri, "Task scheduling algorithms for energy optimization in cloud environment: A comprehensive review," *Cluster Comput.*, vol. 25, no. 2, pp. 1035–1093, Apr. 2022.

[35] V. Mnih et al., "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[36] M. Naphade et al., "The 2019 Ai city challenge," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 452–460.

[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[38] Y. Hui, J. Lien, and X. Lu, “Early experience in benchmarking edge AI processors with object detection workloads,” in *Benchmarking, Measuring, and Optimizing* (Lecture Notes in Computer Science), vol. 12093, W. Gao, J. Zhan, G. Fox, X. Lu, and D. Stanzione, Eds., Cham, Switzerland: Springer, 2020, pp. 32–48, doi: 10.1007/978-3-030-49556-5_3.

[39] Velodyne. *Puck LiDAR Sensor, High-Value Surround Lida-Velodyne LiDAR*. Accessed: Oct. 25, 2022. [Online]. Available: https://velodynelidar.com/products/puck/

[40] Livox. (Oct. 2020). *Livox Avia User Manual*. Accessed: Oct. 25, 2022. [Online]. Available: https://www.livoxtech.com/avia

[41] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, “Wireless networks with RF energy harvesting: A contemporary survey,” *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 757–789, 2nd Quart., 2015.

[42] P. Sunehag et al., “Value-decomposition networks for cooperative multi-agent learning based on team reward,” in *Proc. PMLR*, vol. 80, Stockholm, Sweden, Jul. 2018, pp. 2085–2087.

[43] J. Hribar, L. Hackett, and I. Dusparic, “Deep W-networks: Solving multi-objective optimisation problems with deep reinforcement learning,” in *Proc. 15th Int. Conf. Agents Artif. Intell.*, Lizbon, Portugal, 2023, pp. 17–26.

[44] Z. Wang, Y. Wu, and Q. Niu, “Multi-sensor fusion in automated driving: A survey,” *IEEE Access*, vol. 8, pp. 2847–2868, 2020.

Jernej Hribar (Member, IEEE) received the bachelor’s degree in electrical engineering from the University of Ljubljana, Ljubljana, Slovenia, in 2014, and the Ph.D. degree from the Trinity College Dublin, Dublin, Ireland, in 2020.

From 2020 to 2022, he was a Postdoctoral Researcher with the CONNECT Centre for Future Networks and Communications, Trinity College Dublin. He is a Marie Skłodowska-Curie Action Fellow with the Department of Communication Systems, Jožef Stefan Institute, Ljubljana. His research interests include the age of information, multiagent deep reinforcement learning, and federated learning.

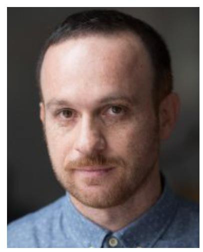
Ryoichi Shinkuma (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in communications engineering from Osaka University, Suita, Osaka, Japan, in 2000, 2001, and 2003, respectively.

He joined the Graduate School of Informatics, Kyoto University, Kyoto, Japan, as an Assistant Professor, from 2003 to 2011, and an Associate Professor from 2011 to 2021. He was a Visiting Scholar with the Wireless Information Network Laboratory, Rutgers University, New Brunswick, NJ, USA, from 2008 to 2009. In 2021, he joined the Faculty of Engineering, Shibaura Institute of Technology, Tokyo, Japan, as a Professor. In May 2023, he founded Hyper Digital Twins (HDT) Company Ltd., Tokyo, where he is currently the CTO. He has published more than 170 peer-reviewed articles. His main research interests include cooperation in heterogeneous networks.

Dr. Shinkuma is a Fellow of IEICE. He received the Young Researchers’ Award in 2006, the Best Tutorial Paper Award of the Communications Society in 2019, and the Best Paper Award from the IEICE in 2022. He also received the Young Scientist Award from Ericsson Japan in 2007, the TELECOM System Technology Award from the Telecommunications Advancement Foundation in 2016, the Best Demo Award Runner-Up at IEEE CCNC 2023, and the Best Poster Award at IEEE CCNC 2024.

Kuon Akiyama received the B.E. degree from the Undergraduate School of Computer Science and Engineering, Shibaura Institute of Technology, Tokyo, Japan, in 2022, and the master’s degree from the Graduate School of Electric Engineering and Computer Science, Shibaura Institute of Technology in 2023. He is currently pursuing the Ph.D. degree with the Graduate School of Functional Control Systems.

He joined Hyper Digital Twins (HDT) Company Ltd., Tokyo, as a Leading Researcher in September 2022. His research interests include smart monitoring with 3-D image sensor networks and its extension for autonomous driving of micromobility vehicles.



George Iosifidis received the Diploma degree in electronics and telecommunications engineering from Greek Air Force Academy, Athens, Greece, in 2000, and the Ph.D. degree from the University of Thessaly, Lamia, Greece, in 2012.

He was an Assistant Professor with the Trinity College Dublin, Dublin, Ireland, from 2016 to 2020, and a Research Scientist with Yale University, New Haven, CT, USA, from 2015 to 2017. He is currently an Associate Professor with Delft University of Technology, Delft, The Netherlands. His research interests lie in the broad area of network optimization and economics. More information can be found at www.FutureNetworksLab.net

Ivana Dusparic received the B.Sc. degree from La Roche College, Pittsburgh, PA, USA, in 2001, and the M.Sc. and Ph.D. degrees from the Trinity College Dublin, Dublin, Ireland, in 2005 and 2010, respectively.

She is a Ussher Associate Professor with the School of Computer Science and Statistics, Trinity College Dublin, a post she took up in 2016. Her research expertise is the development of new artificial intelligence algorithms, specifically reinforcement learning, for the optimization of large-scale infrastructures. She has authored numerous peer-reviewed articles in the areas of reinforcement learning agents, multiagent systems, intelligent mobility, and future communication networks.